1. A method comprising:

initializing to false a predicate that guards a speculative instruction in a software-pipelined loop;

executing at least one iteration of the software-pipelined loop, including an instruction that sets the predicate to true if an associated live-in value is consumed; and

executing the speculative instruction in subsequent iterations of the software-pipelined loop.

2. The method of claim 1, wherein the instruction that sets the predicate true is gated by a stage predicate of the software-pipelined loop.

3. The method of claim 2, wherein executing - at least one iteration of the software-pipelined loop comprises executing the predicate setting instruction when the stage predicate is true.

4. The method of claim 2, wherein the stage predicate is selected to delay execution of the speculative instruction until the live-in value has been consumed.

5. The method of claim 1, wherein initializing to false a predicate comprises initializing to false a predicate other than a stage predicate.

6. A method comprising:

initializing a software-pipelined loop to deactivate a speculative instruction;

executing at least one initiation interval (II) of the software-pipelined loop;

activating the speculative instruction; and

executing subsequent IIs of the software-pipelined loop.

7. The method of claim 6, wherein initializing the software-pipelined loop comprises initializing as false a predicate that guards the speculative instruction.

8. The method of claim 7, wherein executing at least one II of the software-pipelined loop comprises executing an instruction that determines a value for the predicate guarding the speculative instruction.

9. The method of claim 8, wherein activating the speculative instruction comprises executing the speculative instruction if the predicate is true.

10. The method of claim 6, wherein the speculative instruction is a compare instruction and initializing the software pipeline to deactivate the speculative instruction comprises initializing a

rotating source register for the compare to a value for which a predicate determined by the compare instruction is false.

11. The method of claim 10, wherein activating the speculative instruction comprises rotating a value into the source register used by the compare to determine if the predicate is true.

12. The method of claim 7, wherein executing at least one II of the software-pipelined loop comprises executing an instruction that activates the speculative instruction.

13. A method for software pipelining a "while" loop comprising:

identifying a speculative instruction in the loop;

guarding the speculative instruction with a sticky predicate;

initializing the sticky predicate to false; and

inserting an instruction to set the sticky predicate true at a specified initiation interval of the loop.

14. The method of claim 13, wherein inserting an instruction comprises an instruction to set the sticky predicate true when a live-in value targeted by the speculative instruction is consumed.

15. The method of claim 10, wherein the inserted instruction is a compare instruction that is gated by a stage predicate.

16. The method of claim 15, wherein the inserted instruction evaluates the sticky predicate as true when it is gated on by the stage predicate.

17. The method of claim 16, wherein the stage predicate is selected to activate the inserted instruction once the live-in value is consumed.

18. An apparatus comprising a machine readable medium on which are stored instructions that may be executed by a processor to implement a method comprising:

executing a stage of a software-pipelined loop that includes a speculative instruction, the speculative instruction being gated off by a sticky predicate;

executing an instruction that sets the sticky predicate; and

executing the stage of the software-pipelined loop, including executing the speculative instruction.

19. The machine-readable medium of claim 18, wherein the method further comprises initializing the sticky predicate to false to gate the speculative instruction off prior to executing the software-pipelined loop.

20. The machine-readable medium of claim 18, wherein executing an instruction that sets the sticky predicate comprises:

rotating a new value into a stage predicate that guards the sticky predicate setting instruction; and

executing the sticky predicate setting instruction when the stage predicate is true.

21. A computer system comprising:

a processor to execute instructions; and

a memory to store instructions which may be executed by the processor to implement a method comprising:

executing an initiation interval of a software-pipelined loop that includes a speculative instruction, the speculative instruction being gated off by a sticky predicate;

executing an instruction that sets the sticky predicate; and

executing a subsequent initiation interval of the software-pipelined loop, including executing the speculative instruction.

22. The computer system of claim 21, wherein the method further comprises initializing the sticky predicate to false to gate the speculative instruction off prior to executing the software-pipelined loop.

23. The computer system of claim 22, wherein executing an instruction that sets the sticky predicate comprises:

rotating a new value into a stage predicate that guards the sticky predicate setting instruction; and

executing the sticky predicate setting instruction when the stage predicate is true.

24. A computer system comprising:

a processor to execute instructions; and

a memory to store instructions which may be executed by the processor to:

initialize a software-pipelined loop to deactivate a speculative instruction;

execute at least one initiation interval (II) of the software-pipelined loop;

activate the speculative instruction; and

execute subsequent IIs of the software-pipelined loop.

25. The computer system of claim 24, wherein the processor initializes the software-pipelined loop by at least initializing as false a predicate that guards the speculative instruction.

26. The computer system of claim 25, wherein the processor executes at least one II of the software-pipelined loop by at least executing an instruction that determines a value for the predicate guarding the speculative instruction.